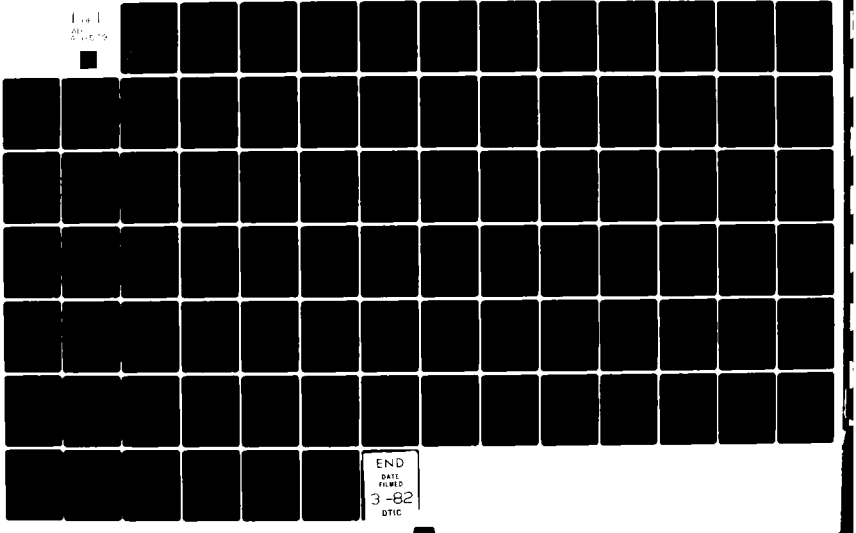
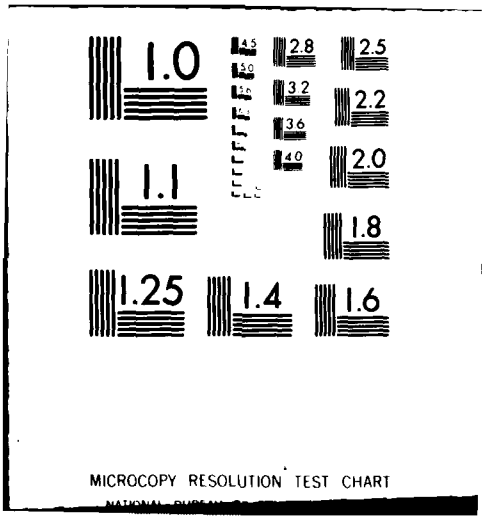
AD-A111 579 FORD AEROSPACE AND COMMUNICATIONS CORP PALO ALTO CA W--ETC F/G 17/2
KSOS SECURE UNIX MAINTENANCE AND SUPPORT PLAN (KERNELIZED SECUR--ETC(U)
DEC 80 MDA903-77-C-0333
UNCLASSIFIED WDL-TR7810-REV-2 NL



END
DATE FILMED
3-82
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART

WDL-TR7810
Revision 2
December 1980

(2)

SECURE MINICOMPUTER OPERATING SYSTEM (KSOS)

SECURE UNIX MAINTENANCE AND SUPPORT PLAN

Department of Defense Kernelized Secure Operating System

Contract MDA 903-77-C-0333
CDRL 0002AE

Prepared for:

Defense Supply Service - Washington
Room 1D245, The Pentagon
Washington, DC 20310

Ford Aerospace &
Communications Corporation
Western Development
Laboratories Division

3939 Fabian Way
Palo Alto, California 94303

Table of Contents

Table of Contents

SECTION 1

INTRODUCTION

The purpose of this document is to present a set of requirements for the successful and secure support and maintainance of the Kernelized Secure Operating System (KSOS). Generally this document is interested in presenting "what" has to be accomplished and not "how" it is to be accomplished. Due to its impact on security issues, the support and maintenance of KSOS presents some interesting and challenging problems. This document discusses in detail the role KSOS plays in overall site application accreditation and, based on this role, the problems associated with satisfying government requirements related to security issues.

Although security issues present the more novel and unusual aspects of KSOS support and maintenance, other issues commonly associated with the support of large software systems are also discussed in detail. For example, problems associated with configuration management and quality assurance of large software systems are discussed and requirements for these disciplines are specified. Of particular interest for KSOS support is the analysis of the different types of potential KSOS sites. Here we are concerned with site categorization based on its security requirements (i.e., how KSOS is planned to be utilized within the site application), special required KSOS customizations, and local site capability (i.e.,

the capability to generate software). It is believed here that the extent of KSOS support and maintenance is highly dependent on the individual sites and their perspective categorizations.

In this document, we hypothesize an organization which will be responsible for providing KSOS support. We refer to this organization as the KSOS Support Organization (KSO). Throughout this document we define, analyze and specify requirements for the interaction between the KSO and potential KSOS-based application sites.

The remainder of this section will present brief background information on KSOS and provide a short synopsis of the remaining six sections of this document.

1.1 KSOS BACKGROUND

The Kernelized Secure Operating System (KSOS) was developed to provide a secure means to operate in a multi-level security environment. It has satisfied all requirements of the governmental security control agencies for handling different levels of secure data without compromise.

The system was designed to emulate the UNIX operating system and at the same time provide the integrity of classified data. It is divided into three basic parts: 1) the security kernel, 2) UNIX emulator, and 3) non-kernel security-related software.

The KSOS Security Kernel supports the emulation of the standard UNIX operating system within the constraints of the multi-level security model. The Security Kernel provides all the data and computational objects required to construct a general purpose operating system. The Security Kernel mediates all information exchanges within the system permitting only those exchanges which are consistent with the multi-level security model. The Security Kernel provides secure information channels which permit the user to communicate directly with "trusted" software processes. The secure information channels are not subject to compromise due to "spoofing" by untrusted software.

The UNIX Emulator creates a system call interface which is compatible with that provided by the UNIX operating system. The UNIX Emulator does this by mapping the user's UNIX system calls into the appropriate sequences of Security Kernel calls. The Emulator creates the hierarchical UNIX file structure from the more primitive structure supported by the Security Kernel. The Emulator also contains the per user aspects of support for the interface to a computer network.

The Non-Kernel Security-Related Software (NKSRS) is a collection of autonomous subsystems some of which execute with extraordinary privilege (i.e., more privilege than is afforded to user programs) to provide essential services to the system. Such services include, but are not limited to, the following:

a. Secure User Services, those services invoked by users which must have a trusted path to the service, such as the login process.

b. System Operation Services, those functions essential to the operation of a KSOS system, such as the Network Daemon.

c. System Maintenance Services, those functions needed for continued operation and maintenance of a KSOS system, such as dump and restore of file systems.

d. System Administrator Services, those functions needed to support the administrative operation of the system, such as the adding and deleting of users.

1.2 DOCUMENT ORGANIZATION

The remainder of this document is divided into six sections. Section 2 deals with the KSOS support and maintenance problem in general. Here the maintenance problem is discussed with respect to overall security issues including the problem of site application accreditation. Problems concerning the distribution/redistribution of KSOS and discrepancy report (DR) control and resolution are discussed in his section. Conventional problems dealing with on-site training, configuration management, and quality assurance are outlined in this section. Also in Section 2 is a short discussion on the different software components of KSOS from the point of view of support and maintenance responsibilities. The

definition of "KSOS" presented in this section is important from a support and maintenance point of view.

Section 3 presents the structure of the KSO. Here four functional departments are briefly discussed, each having responsibilities in the areas of configuration management, quality assurance, software design and development, and verification. Short scenarios are presented showing the interplay between KSO departments for various aspects of KSO operation.

Section 4 presents a detailed discussion of KSOS application sites. Here sites are analyzed with respect to the characteristics affecting KSOS support and maintenance. In particular, site security requirements of KSOS, availability of site personnel for KSOS operation, special site customization requirements, and local on-site capabilities for producing software are discussed.

Section 5 specifies the requirements of the KSO. The four departments and approval board outlined in Section 3 are defined in detail in this section. This section also lists the set of existing support and maintenance facilities currently available for continued support and development.

Section 6 specifies the responsibilities of individual KSOS sites. These include overall government approval for KSOS utilization with classified information, specification of site categories for the KSO, identification of qualified personnel for KSOS operation, KSOS/personnel training, discrepancy report (DR) handling, and management of special on-site capa-

bilities.

Section 7 presents a set of scenarios illustrating the interplay between the KSO and KSOS sites. These scenarios cover situations involving unclassified sites, classified sites operating "system high", classified sites operating in a multi-level environment, KSO/site interchange for trusted software development, and KSO/site interplay for DR resolution.

SECTION 2

THE KSOS MAINTENANCE PROBLEM

In this section we discuss the problems involved with the overall support and maintenance of a security kernel based operating system as exemplified by KSOS. Although problems associated with the security aspect of KSOS are the most unusual and present the most new challenges in software support, other more typical problems such as configuration management or discrepancy reporting are also present.

The following KSOS support and maintenance problem areas have been identified:

a. Overall security issues

b. Accreditation of KSOS-based applications

c. KSOS distribution/redistribution procedures

d. Discrepancy Report (DR) procedures and control

e. KSOS 'customization' developments

f. KSOS training and operation

g. Configuration Management and Quality Assurance

Before presenting a general overall of the KSOS-related problems listed above, it is prudent first to establish a definition of KSOS software components in a context applicable to this document. This definition and problem discussions now follow.

2.1 KSOS SOFTWARE COMPONENTS

The purpose of this subsection is to define precisely those KSOS software components which are applicable to this document. In general, this document must specify the responsibilities of the KSOS Support Organization (KSO) for the distribution and overall support of KSOS. In order to accomplish this goal, it is important that the KSOS system be broken into those components which are under the control of KSO with respect to both system security and system integrity.

Figure 2-1 presents a simplistic view of the software components comprising a KSOS-based system. The three concentric circles represent the three operating domains of the KSOS machine; the kernel, supervisor and user domains. The most inner circle contains the KSOS security kernel, the second circle contains the UNIX emulation package, Non-Kernel Security Relevant Software (NKSRS), and potential site-specific trusted software, and finally, the outer circle contains site-specific application software.

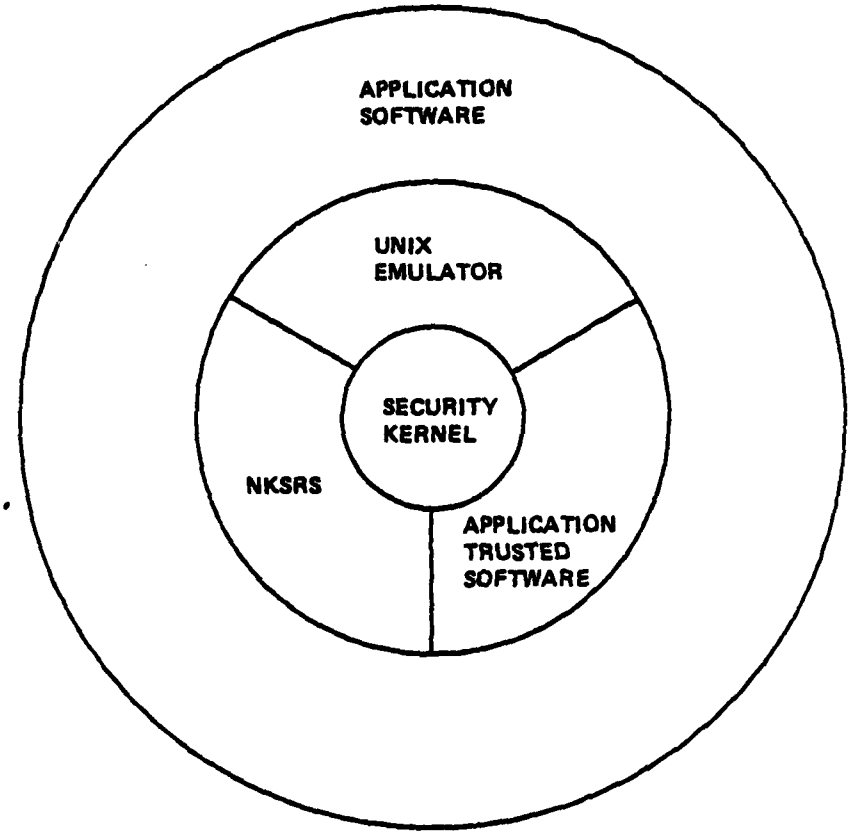


Figure 2-1. KSOS Software Components

For general distribution of KSOS to new sites,the following software components comprise the "standard" KSOS system:

a. Security Kernel

b. UNIX Emulation Package

c. NKSRS

Based on specific KSOS site requirements, application software must be supplied and/or developed. Normally, this is the sole responsibility of the site. However, if site requirements dictate that new "trusted software" be developed, then the KSO must be involved.

KSO involvement is required for two reasons:

a. Trusted software is "security relevant" and, therefore, must be accredited by KSO prior to operational use.

b. Trusted software development and verification involves special expertises and tools normally only available to KSO.

The first reason is mandatory due to KSO's important role in the overall accreditation of a sites operation involving multi-level access of sensitive information. The second reason is one of practicability. Generally, the development of trusted software involves special knowledge of the Security Kernel and expertise in overall secure system generation. Although a site may have this expertise, it will be the exception, not the rule. Even if a site did have government and/or contractor personnel

conversant with KSOS and secure system development, the actual verification involving formal specifications, potential "code proofs", etc. is a costly process involving access to special verification tools.

It is important to point out here that site-specific application trusted software is closely related to the NKSRS package. The difference here is that NKSRS is a 'standard' KSOS package that is entirely under the control of the KSO. Possibly, if some site-specific trusted software turned out to be required at many sites, it could be absorbed within the NKSRS package. However, what ever the case, site-specific trusted software in Figure 2-1 depicts NEW trusted software requirements of a site destined for KSOS.

KSOS software components depicted in Figure 2-1 which are security relevant are the following:

a. Security Kernel

b. NKSRS

c. Site-specific Application Trusted Software

KSOS software components whose maintenance and support responsibilities are under the KSO are:

a. Security Kernel

b. NKSRS

c. UNIX Emulation Package

d. Site-specific Application Trusted Software

It is interesting to point out here that the trusted software is included in the second list above due to security reasons, not for overall integrity purposes.

In summary, the standard KSOS software set is comprised of the security kernel, UNIX emulation package, and the NKSRS. However, from the point of view of KSO responsibility, the standard KSOS package plus potential site-specific trusted software must be considered as a package. Therefore, in the remainder of this document KSOS will refer to both the standard KSOS package and potential site-specific trusted software. The distinctions brought out above will prove to be useful in later sections of this document when KSO and KSOS site responsibilities are defined.

2.2 SECURITY

The very name of KSOS implies a security sensitivity not found in any other operating system. Many of the fundamental technologies comprising KSOS development are more concerned with the preciseness of specification and correctoness of operation than security per se. It is well recognized that any algorithm which guarantees "zero fault" operation must be developed along the same lines as KSOS. However, it is the "security nature" of KSOS that demands correctness of operation. Proving correctness of operation (i.e., verification) is essential for guaran-

teeing that KSOS can safely support a multi-level environment.

KSOS is, as the name applies, based on security kernel technology. Security kernel technology can be summarized essentially as follows. A security kernel must satisfy three attributes:

a. It performs its tasks (and NO other tasks) correctly.

b. It is protected.

c. All access attempts are controlled by the kernel.

The first attribute implies that the security kernel must go through some type of formal verification to "prove" that it does indeed operate correctly. The second attribute requires that the kernel be protected (by hardware) from other executing programs. The third attribute implies that all computational resources are controlled by the kernel (e.g., memory management, interrupt processing, devices, etc.).

Although the security kernel may satisfy the three fundamental attributes discussed above, in order for it to provide a 'multi-level' environment it must also implement a "security model" of some sort. KSOS implements a "simplified" version of the DoD Security Model. This security model is based primarily on the mathematical model developed by Bell and LaPadula. In particular this model entertains the standard DoD security level and compartment distinctions commonly found in in DoD: Security levels are unclassified, confidential, secret, and top secret and they abide by strict relational operations (e.g.,secret is "higher" than confidential,

etc.). Compartments are sets of security caveats (e.g., SI, NOFORN, etc.). Generally, the KSOS simplified DoD Security Model dictates that an "object-a" can reference (read) another "object-b" if and only if object-a's security level is greater-than-or-equal to object-b and object-b's compartment set is contained within object-a's compartment set. It also dictates that an object-a can update (write) object-b if and only if its security level is less-than-or-equal to that of object-b and its compartment set is contained within object-b's compartment set. This can be stated simply as objects can "read low" and "write high", where low and high refer to classification levels/compartment set distinctions.

Therefore, a security kernel which conforms to the three attributes of correctness, protection, and completeness and implements a DoD Security Model as described above, constitutes the minimal requirements for enforcing a multi-level secure environment.

The "security oriented" nature of KSOS introduces some unusual problems associated with the maintenance and support of the system. These problems can be divided into two groups:

a. The security sensitivity of the sites and applications utilizing KSOS.

b. The verification/reverification procedures to guarantee KSOS conforms to the "correctness" attribute imposed on security kernels.

The first problem of "site security sensitivity" is not caused by KSOS per se, but is in fact site dependent. Here we are referring to the KSOS-based applications (i.e., application software), the classification level of the data passing through the application and the functions and operations made on this data. Typically, these applications will be UNIX-based software packages performing specific user tasks; e.g., terminal concentrators for access to secure networks, message systems, sanitization/downgrading mechanisms, etc. Security sensitivity of a site is dependent on nature of data classification and the operations made on this data. For example, the system high classification level of the data (e.g., is it secret or top secret, etc.) or is the application operating in a multi-level environment and if it is, what is the security level span (e.g., confidential-to-secret, secret-to-top secret/SI, etc.), are all issues on security sensitivity of a site. The operations made on classified data poses an interesting issue. Here we are referring to operations which, for overall application requirements, may circumvent the security principles of KSOS itself. For example, controlled santization and downgrading of sensitive information. This type of application may violate one or more of the DoD Security Model principles. Of course, the man/software interaction to accomplish this function must be carefully controlled, the man requiring overall cognizance of security issues and the sensitivity of data, and the software of a "trusted" nature.

In summary, site security sensitivity is based on the following attributes of site application operations:

a. Security level/compartment set of application operation.

b. If the application is multi-level, what is the security level/ compartment set range it operates within.

c. Special controlled operations circumventing KSOS security protection.

The second problem is definitely the most difficult and epitomizes the unusual nature of KSOS. As discussed earlier in this section, one of the attributes of a security kernel is that it performs its operations (and only those operations) correctly. The "proof" part requirement of this attribute involves a set of non-trivial processes and procedures that must be applied to KSOS. All "security relevant" components of KSOS must go through this "proof" or verification process. As discussed above, security relevant components of KSOS are the security kernel itself, NKSRS, and potential site-specific trusted software. All of these components must go through stringent verification processes in order to "guarantee" their correct operation. These processes involve formal specifications derived from the "English specifications", proofs that the formal specifications satisfy some security and/or operational requirement, special programming requirements involving strong "type checking" languages, and potential "code proofs" of the software itself. The processes above are time consuming laborious tasks which must be re-initiated each time a security relevant component is altered. This verification/reverificaton process is definitely the most difficult problem confronting the support and maintenance of KSOS. The problem becomes

even more involved when verification requirements are integrated with site application security categories and overall accreditation issues.

2.3 ACCREDITATION OF KSOS-BASED APPLICATIONS

The accreditation of any security sensitive operation is a lengthy and complicated process. In order for the government to accredit the operation of some site many issues must be resolved. These issues are concerned with site physical security, personnel security (i.e., security clearances), communication security (i.e., encryption/decryption), TEMPEST considerations, specific agency approval of data access, etc. The insertion of a KSOS-based system into this accreditation process introduces yet another facet for overall government approval. It is important to stress here, however, that KSOS is only a part of this accreditation process. Typically, a site which requires the use of a KSOS, which would normally imply that the site must operate in a multi-level environment, must gain initial approval of its overall operation (which KSOS is only a part).

It is believed here that the organization that is responsible for the distribution, support and maintenance of KSOS (the KSO) must itself be "accredited" by the government to perform these duties. If this is not the case, then the overall accreditation of any site which uses KSOS will not be possible. In order for KSO to be accredited, its operations and procedures for supporting KSOS must be fully critiqued by the cognizant government agency to guarantee that they do indeed comply to standard

security policies.

2.4 KSOS DISTRIBUTION/REDISTRIBUTION

The KSO will be responsible for distributing "standard" KSOS packages to those sites approved by the government. The problems involved with fully supporting any software system which is located in many sites each possibly having special versions or packages, is also present with KSOS support. From a pure procedures point of view, the distribution/redistribution of KSOS is not any different than any other fully supported software system. However, the security nature of KSOS does present some additional problems. Here we are referring to the redistribution of KSOS systems which have "fixes" for one or more security relevant components. For example, if a KSOS-based site operating in a multi-level environment encounters a problem which may indicate "an error" in some security relevant component of KSOS, then KSO may be forced, depending on the severity of the problem, to "shut down" all other sites containing this component. This example is probably more realistic where special trusted software is being utilized. Here, if the trusted software is site-specific, then the problem is localized to that site, hence not affecting other KSOS-based sites.

The more usual problems associated with distribution/redistribution are those concerned with software maintenance and support procedures which address the proper methods of tracking standard system releases, new releases, old releasrs, special customized system derivatives, etc.

Since KSOS is a software system, all of these procedures are required by the KSO.

2.5 DISCREPANCY REPORT (DR) PROCEDURES/CONTROL

Key to the successful maintenance and support of any software system which is distributed across a set of sites are the procedures and controls for discrepancy reporting (DR). DR procedures involve the establishment of a DR form to be distributed at the various KSOS sites, training of site personnel in using the DR form, proper control communication channels for DR submittal to KSO, sufficient priority classifications for DR rectification, and overall procedures for configuration management/quality assurance operations of the total DR sequence.

Due to the security nature of KSOS-based sites, additional problems may occur with DR procedures. For example, if in order to properly convey a probable discrepancy to the KSO, the site must also include operational data which is sensitive, then KSO must establish an official channel for the safe transmittal of that information. This of course also implies that KSO personnel must have the appropriate clearances to handle this information. It is believed in this document that in order for KSO to fully satisfy the large task of supporting KSOS and its potential sites, its personnel and facilities must have the appropriate clearances to handle sensitive information.

2.6 KSOS ENHANCEMENT MODIFICATION

The KSO will be requested by some potential KSOS sites to incorporate enhancments into the KSOS package. These enhancements would be normally concerned with alterations to the UNIX emulation package or possibly new trusted software. It is not expected that requests for alterations to the security kernel itself would be an unusual occurrence.

KSOS enhancement requests may occur prior to the initial distribution of a KSOS system to a new site, or may occur after the site has been using KSOS for some time. In either situation, KSO must have sufficient qualified personnel to develop these enhancements. It is important to stress here that the KSO should exploit on-site capabilities whenever possible. Some potential KSOS sites will have considerable on-site capability in order to develop partial and/or all required enhancements. KSO must be able to interface with the sites accordingly, accommodating those portions only it can perform. In particular, trusted software development involves many special expertises that will not be available at most sites. Trusted software development probably is the most important and difficult aspect of "KSOS customization" that the KSO must provide. First, most KSOS-based applications will involve one or more trusted software packages. This is particularly true for sites utilizing existing software packages (e.g., UNIX application packages) for non-multi-level environments where a need for multi-level operation has come about. Trusted software development is difficult for several reasons. First, it requires expertise in writing software interfacing directly to the secu-

rity kernel. The UNIX emulator interface cannot be utilized (it is not secure). Second, since trusted software must be proved to operate correctly, it must go through special verification processes, a time consumming and costly procedure. Third, trusted software must go through special "handling" procedures in its distribution and subsequent support. It is believed that the support of KSOS "customization" will become a significant portion of KSO duties.

2.7 KSOS SITE TRAINING

Since KSO will be responsible for both the security and integrity of KSOS operation, it is important that individual sites be trained in the day in and day out operations of KSOS. KSOS site training should be a standard part of any KSOS pre-delivery phase.

Training will be concerned with both security and integrity issues. Typical security issues are the secure disposition of the security sensitive software, secure system loading procedures, appropriate doctrination of site security officers and cognizant managers of the KSOS-based application, etc. Integrity issues are the appropriate training for DR submittals, proper file system backup and recovery procedures, etc.

Special training may also be required for those sites desiring to make approved alterations to the KSOS system. Typically we are referring to non-security relevant software enhancements.

In summary KSOS site training will ensure that the site is prepared to operate their KSOS-based application in a secure and reliable manner. Since KSO has overall responsibility for the security and integrity of KSOS, site training is an important and necessary component in the overall maintenance and support of KSOS.

2.8 CONFIGURATION MANAGEMENT/QUALITY ASSURANCE

Configuration management and quality assurance is crucial to the successful operation of KSO since it in a sense guarantees the "correctness of operation" of KSO in performing the many duties discussed above. Here configuration management is responsible for ensuring that sufficient procedures are defined and continuously followed throughout the evolution of KSOS development. This includes procedures for tracking standard KSOS releases and special "derivative" KSOS systems, receipt and proper processing of DRs, and the factoring of DRs into subsequent new releases.

Quality assurance is a parellel discipline to configuration management whose responsibility is to ensure that all KSOS enhancement requests are reviewed for soundness (with respect to system integrity and security), that they are developed correctly, and tested thoroughly before handed over to configuration management. Quality assurance will also be responsible for guaranteeing that security relevant components of KSOS are maintained, enhanced and developed with the same rigor and thoroughness as the original system. This includes all formal verification processes, such as formal specifications, specifciation proofs, and code proofs.

SECTION 3

KSOS SUPPORT ORGANIZATION (KSO)

This section describes the basic structure of the KSOS Support Organization (KSO). Here a general overview of KSO suborganizations, termed departments, is presented showing the interplay between these departments in providing services to the KSOS site community.

Section 5 will provide a detailed description of the requirements of KSO as outlined in this section.

3.1 KSO STRUCTURE

Figure 3-1 shows the basic suborganizations within the KSO. The KSO is managed by a director who has four managers reporting to him. Each manager is in charge of one functional suborganization of KSO. These are:

a. Quality Assurance Department

b. Configuration Management Department

c. Software Design and Development Department

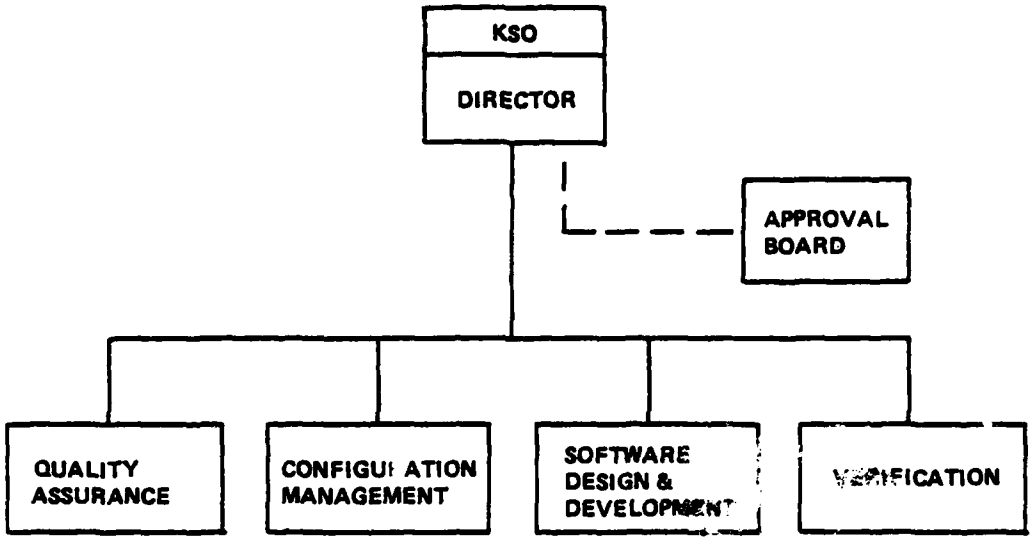


Figure 3-1. KSOS Support Organization (KSO) Structure

d. Verification Department

The Quality Assurance Department of KSO is responsible for ensuring the overall quality of KSOS support and maintenance. This department must always approve all alterations to KSOS, as well as conduct periodic checks on KSOS sites to guarantee total compliance to KSOS operational procedures. Within KSO, Quality Assurance guarantees the "correctness" of operation. This includes complete knowledge and oompliance to government and military standards, software enhancement development and verification, discrepancy report tracking and rectifications, and proper on-site training and software installation procedures.

The Configuration Management Department is responsible primarily for controlling all KSOS software and KSOS related material. The Configuration Management Department is responsible for tracking all discrepancy reports (DR), maintaining up-to-date KSOS documentation, on-site training, and KSOS installation.

The Software Design and Development Department is responsible for developing all software enhancements to KSOS. Personnel in this department will have intimate familarity with all KSOS components and will be capable of developing customized trusted software for those KSOS sites requiring it. In addition, this department will be responsible for resolving all DRs received from KSOS sites.

The Verification Department will be responsible for all verification/reverification tasks required of KSO. This will include the

capability for producing high level formal specifications for security relevant software, low level formal specifications, conducting proofs of formal specifications and "code". This department will have intimate familiarity with existing formal specifications and proof techniques currently used for the development of KSOS.

Finally, KSO has an Approval Board chaired by the Quality Assurance manager. This board is composed by the KSO director, and the four KSO department managers. This board's primary responsibility is to guarantee the overall integrity of the charter of KSOS during its evolution. In particular, all KSOS enhancement requests must be approved by the KSO Approval Board prior to development by other departments of KSO. KSOS site approval is also given by this board, who must interface with the appropriate government agencies to gain permission for KSOS distribution. New trusted software must be approved by this board to ensure that it does indeed comply to the strict security and integrity standards of the government.

3.2 KSO DEPARTMENT INTERPLAY

Before going into detailed descriptions of the tasks performed by the various KSO departments, an overview of KSO department interaction is appropriate. Figures 3-2 through 3-4 illustrate the order of KSO department action based on three tasks:

a. UNIX emulator enhancement request

b. Special Trusted Software request

c. Receipt of a DR

Assume that a potential KSOS site has requested that new system calls be added to the existing UNIX emulation package (Figure 3-2). This request is first reviewed by the KSO Approval Board. The Approval Board analyzes the request insuring that the enhancements are sound and implementable. Once the request has been approved by the Approval Board, it is turned over to the Software Design and Development Department of KSO. This department will be responsible for the entire design, implementation and checkout of the UNIX enhancements. Following successful development of the new UNIX system calls, the Quality Assurance Department of KSO must verify that the newly developed software enhancements do indeed comply with the original site request. This involves checking carefully the tests run against the new system calls making sure that they do perform correctly. Finally, following Quality Assurance approval, the new UNIX emulator enhancements are given to the Configuration Management Department for insertion into KSOS for delivery. Configuration Management inserts the new software into the appropriate "source chain" for subsequent delivery to the site. Appropriate documentation updates are also generated by the Configuration Management Department reflecting the new UNIX emulator system calls.

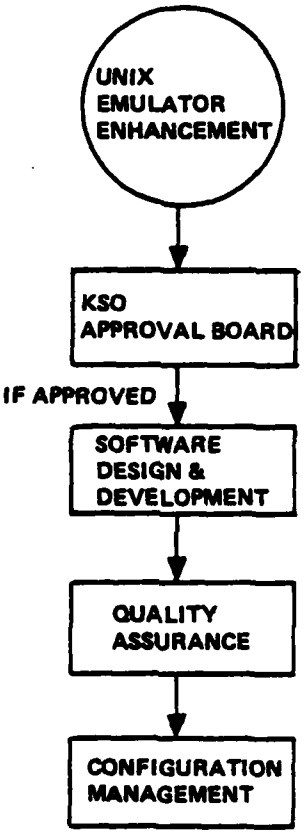


Figure 3-2. UNIX Emulator Enhancement Request

Figure 3-3 illustrates the situation where a request for new trusted software is received from some potential KSOS site. This situation is more complicated than the previous one. Again, as with all KSOS change requests, the KSO Approval Board must review and approve of the requested new trusted software. Here, the Approval Board will make a judgement as to the soundness of the trusted software and its inherent complexity. If the Approval Board decides that the trusted software functionally, as defined, is too complex to allow verification, then it will have to be simplified or turned down all together. Assuming that the Approval Board accepts the proposed new trusted software, then the Verification Department of KSO is given the request for formal specification. This involves taking the "English" specification of the trusted software and formally specifying it in a specification language. This formal specification is then given to the Software Design and Development Department for implementation. The formal specification will form the design baseline for the trusted software. In parallel with the development phase, the Verification Department will proceed to prove the formal specifications. Following successful specification proofs and trusted software development, the Verification Department will then verify the actual programs comprising the trusted software. Depending on what the trusted software does, actual "code proofs" may be necessary for verification. The Quality Assurance Department then reviews the entire set of sequences and tests used in developing and verifying the trusted software to ensure that the end product was developed and tested properly. Following Quality Assurance approval, the Configuration Management Department is given

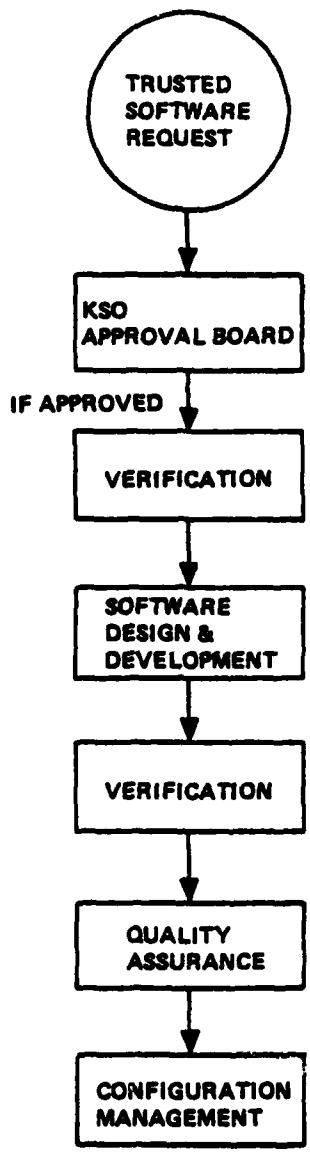


Figure 3-3. New Trusted Software Request

the trusted software for insertion into the appropriate KSOS system "source chain".

Figure 3-4 illustrates the chain-of-command for the processing of Discrepancy Reports (DR). As with all KSO requests, the KSO Approval Board reviews all DRs submitted to KSO. If the DR is potentially security relevant, then the Approval Board will notify all affected sites of the problem. In either case, the DR is then given to the Configuration Management Department for proper registration. The Configuration Management Department assigns a priority to the DR based on the priority of affected sites. If the DR affects security relevant software, then the Verification Department is given control of DR resolution. The Verification Department will analyze the problem determining if it requires respecification. If it does, the Verification Department then respecifies that portion of affected functionality and submits the new specification to the Software Design and Development Department. This department then proceeds to modify the software according to the new specifications. Also, if the problem required new specifications, the Verification Department must re-initiate those specification proofs affected by the change. Following required reverification and software development, the Verification Department again inspects the software for compliance to the new specifications and conducts new "code proofs" if required. At this point, Quality Assurance then reviews and approves of the corrected software and passes it on to Configuration Management for dissemination to the appropriate KSOS sites.

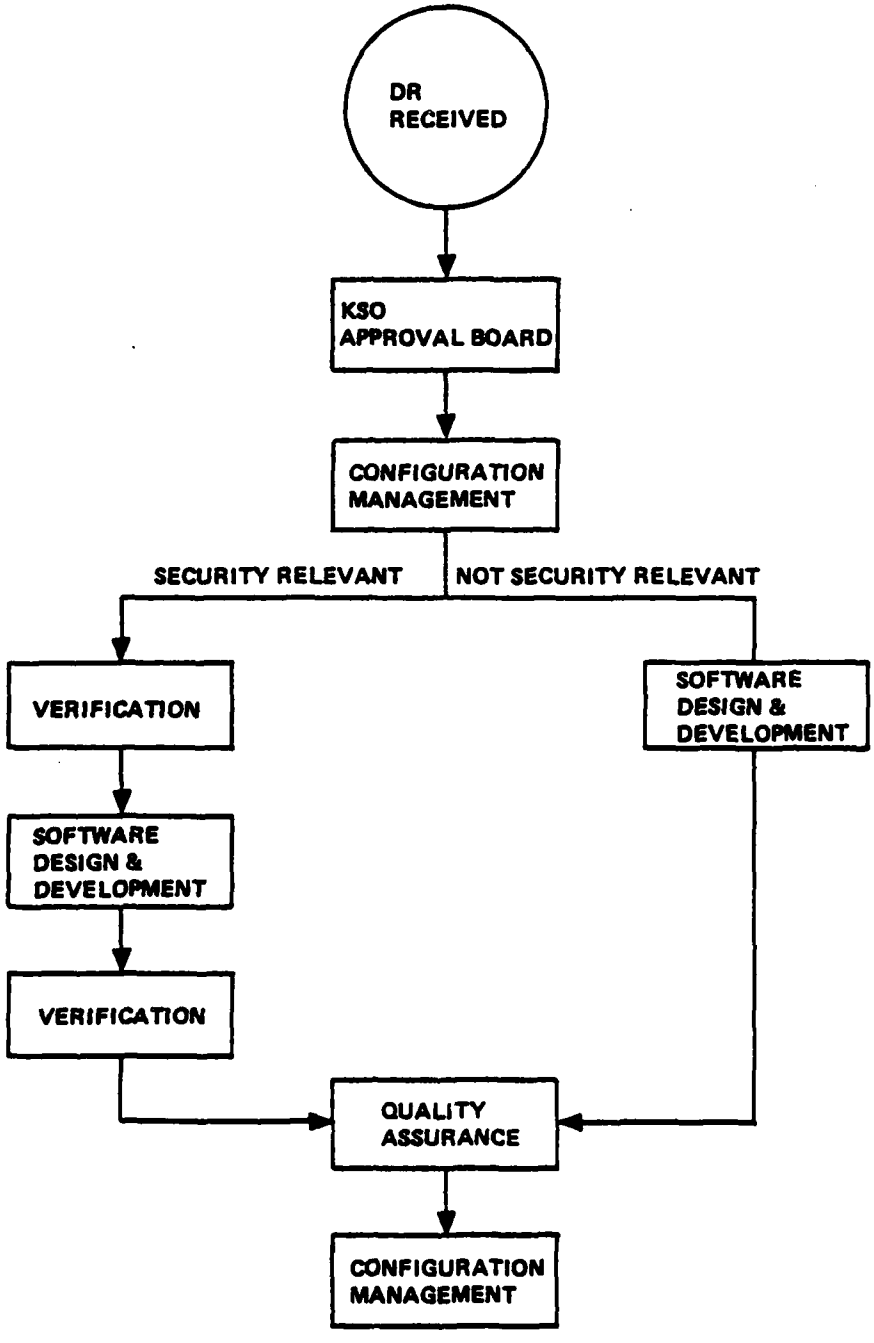


Figure 3-4. Discrepancy Report (DR) Processing

If the DR did not affect security relevant software, then the Verification department would not be involved with its resolution.

SECTION 4

KSOS USER SITES

One of the key considerations in specifying the functions of the KSOS Support Organization (KSO) is the categorization of potential KSOS user sites. KSOS user site categorization must be determined on a number of site characteristics which can potentially affect the responsibilities of KSO. These characteristics are:

a. site security

b. site personnel

c. special site requirements

d. site capabilities

4.1 SITE SECURITY

Site security is a major item in categorizing potential KSOS user sites. Generally, by site security we are referring to the security level the KSOS system will run under as an operational system. We are interested not only in the highest level it operates under, which is important, but the range of classification levels and compartments required. It is believed that potential KSOS user sites will span all combinations of

system high and classification level ranges (multi-level operations). The determination of these security characteristics dictate the scope of responsibility of the KSO.

For example, a potential KSOS user site may wish to operate in a Secret-to-Top Secret multi-level environment. Another system may not require multi-level operation and will only operate in a system high mode (e.g., Secret). Yet another may require multi-level operation plus special privileges to allow controlled downgrading of information. Depending on the security categorization of a potential KSOS user site, the extent of KSO participation is directly affected for overall site accreditation, priority of Discrepancy Report (DR) processing, etc.

Following is a set of security categorizations that will affect the scope of KSO responsibilities in supporting a KSOS user site:

a. unclassified

b. System high, confidential, or secret, so on.

c. Multi-level, unclassified-to-secret, unclassified-to-top secret, and so on.

d. Multi-level as in c, plus special controlled privileges (e.g., downgrading, level changing, etc.).

The first security category (unclassified) would typically be the case where some university desired a copy of KSOS for research projects. From a security view point KSO responsibilities would be nil for this

situation. The next category, operating in system high (a non multi-level environment), would probably not represent a common occurrence since the primary purpose of KSOS is to provide multi-level security (it is included here for completeness). What is interesting to note here is that if a problem did occur with a KSOS-based application operating in a "system high" site, this wouldn't cause problems that are security relevant. In other words, from a security point of view, operating KSOS in a system high mode is an identical situation as using any other operating system.

Once a KSOS user site becomes involved with multi-level operation, security categorization becomes important with respect to KSO responsibilities. Here both the range of security levels and the highest level are important categories. For example, if a system flaw is discovered in KSOS it is, from a security view point, more serious for a user site operating a multi-level operation of confidential-to-top secret, than a user site operating unclassified-to-confidential. Here the sensitive issue is the number of levels spanned, not necessarily the highest level involved.

Finally, the most critical security category is a KSOS user site operating in a multi-level environment where one or more special privileges are available for circumventing certain properites of the DoD Security Model. It is interesting to point out here that the first designated application for KSOS, ACCAT GUARD, is of this critical security category. Not only does this KSOS user site operate in a multi-level environment of

secret-to-top secret SI/SAO, it is privileged to allow the controlled "writing down" (called "downgrading" in GUARD) of sanitized information.

4.2 SITE PERSONNEL

In order for KSO to approve a potential user site for KSOS operation, a designated set of on-site personnel must be available and trained for installation and operation of the system. In most situations it is believed that these individuals would already exist at most sites which handle classified information and operate large EDP centers. However, certain on-site responsibilities must be delegated inorder to operate a KSOS-based system in a sound and secure manner. The following on-site personnel are identified:

a. Security Officer (SO)

b. System Administrator (SA)

c. System Operator (SYO)

The Security Officer (SO) will normally already exist on any site which processes sensitive classified information. The on-site SO will have overall security responsibilities of site physical security, personnel security (i.e., individual clearances), and the enforcement of basic security policies and procedures as dictated by the government. With the addition of a KSOS-based system operating in a multi-level environment, the SO's responsibilities will be increased to include interfacing with the KSO. These additional responsibilities will be the acknowledgement

of necessary KSO personnel, physical confinement procedures for KSOS operation, etc. In many cases, the arrival of a KSOS-based system will be similar to any computer system already operating at the site. However, the potential multi-level nature of the system will require additional responsibilities of the SO.

The System Administrator (SA) will have personal responsibility for the day in and day out operational procedures of the KSOS-based system. This individual will be responsible for the secure disposition of the KSOS object tapes/disks, submission of KSOS discrepancy reports, supervision of the KSOS System Operator(s) (SYO), etc. Typically, this individual would have overall management of the application that was utilzing KSOS, and hence would already exist at the site. As with the SO, the SA's responsibilities would be increased to cover the items mentioned above.

The System Operator (SYO) will provide the everyday tasks of bringing the KSOS-based system up, performing appropriate file system backups, and other duties related to system operation. The SYO will report to or someone delegated by, the SA.

4.3 SPECIAL SITE REQUIREMENTS

Special site requirements constitute those special situations where non-standard hardware devices may be employed, new site-specific trusted software is needed, additional and/or altered UNIX emulator system calls are required, etc. All of these directly affect the overall KSO installation and support efforts.

For example, a potential KSOS user site may have purchased a non-standard DEC disk, or possibly a DEC disk which isn't supported by the present KSOS system. For either situation, a new device driver must be generated for this site. This particular example is interesting for a number of reasons. Since all device drivers reside in the KSOS Security Kernel itself, this involves system modifications in areas that are potentially security relevant (i.e., any change to the security kernel is security sensitive). Since availability of the UNIX emulation package in KSOS will attract potential user sites who have existing UNIX-based software, the request for UNIX emulator changes will not be an infrequent occurrence. It is well known that many "dialects" of UNIX exist in industry today. Examples of UNIX emulator changes are: provision for a UNIX level exclusive open system call, provision for a different Inter-Process Communication (IPC) mechanism, etc.

The requirement for new site-specific application trusted software is another site requirement which will probably prove to occur quite frequently. This requirement is especially interesting and will probably prove to be KSO's most challenging job. Trusted software is typically that software which is given privileges to "violate" one or more of the DoD Security Model principles. Typical examples are the "writing down" of information or "level changing" of information "containers". Other related trusted software is that which can grant and invoke privileges, manipulate integrity privileges, etc.

The reason this special software is called trusted is because its operation must be guaranteed to operate correctly. In order to provide this guarantee, its operation must be carefully specified and its subsequent implementation must follow the specification precisely. Currently with the KSOS development, security relevant software is formally specified with a specification language which is then verified to satisfy its required security principles.

Trusted software involves implementation considerations not normally present with typical application software development (i.e., UNIX-based software). Since the trusted software, in order to guarantee its correctness of operation, must utilize only that software which is formally verified (i.e., the security kernel or other trusted software), it cannot be implemented within a UNIX environment. The KSOS Unix emulation package is not verified to operate correctly. Therefore, trusted software must be implemented using direct system calls to the security kernel itself (called kernel calls). Secondly, since potential code proofs may be required of the trusted software for its certification, the Modula programming language must be utilized. Modula possess the strong "type checking" characteristics required for code proofs.

It is clear from the above discussion that trusted software involves potentially four phases that do not normally occur with the development of typical application software:

a. Formal specification of its operations.

b. Operating system environment must be the Security Kernel.

c. Programming language must be Modula.

d. Potential requirement for verification of formal specifications and code.

It is believed here that those KSOS user sites which have significant expertise in developing large UNIX-based systems will probably not be conversant enough with KSOS to generate trusted software. The formal specification phase of this task alone would exclude most sites.

In later sections we will discuss the interplay between the KSO and a hypothetical KSOS user site involving the development of trusted software.

A site which doesn't have special requirements would be those sites which only required the standard KSOS package (sans any site-specific trusted software). It is important to note here that if a site had the appropriate software development capability (see next subsection) then some of the special tailored requirements could be absorbed by the site directly, hence not increasing KSO responsibilities.

4.4 SITE CAPABILITIES

KSOS user site capabilities include those expertises that a particular site may have available for developing software germane to the KSOS application requirements. In general it is safe to assume that most candidate KSOS sites have some particular application that requires the secure formalism provided by KSOS (i.e., some type of multi-level operation). Based on this application, it will be normally the case that either on-site government personnel and/or designated contractors will be responsible for the entire development of the application. Depending on the special requirements (see previous subsection) of the site and the software capabilities available, the scope of the support effort of KSO is affected accordingly. It is envisioned here that the KSO would be able handle on a need-by basis any type of software development short of producing actual site-specific application software. Even with this caveat, there is the exception of site-specific trusted software. For this case, KSO would be the probable developer. As with the example of trusted software development presented in the previous subsection, KSO and site interplay becomes somewhat complicated. Typically one would expect the site to state operational requirements of the trusted software and then the KSO would develop the actual software, including formal specifications and verification of these specifications. Even if the site had the special expertise to perform this special tasks, KSO would still have the final accreditation responsibility for the trusted software.

Therefore it is clear from the above discussion that the amount of additional support that KSO may be required to supply is tied directly to the on-site software capabilities available.

SECTION 5

KSOS SUPPORT ORGANIZATION (KSO) RESPONSIBILITIES

In Section 3 we presented an organizational overview of the KSOS Support Organization (KSO), identifying four functional departments and a committee. In this section we will define the responsibilities of each KSO functional area in detail. As brought out in other sections of this document, characteristics of given KSOS user sites determine the extent of responsibilities of KSO in its overall KSOS support role. Section 6 presents KSOS user site responsibilities, which when combined with KSO responsibilities define the complete KSOS support scenario.

As initially defined in section 3, KSO is made up of one approval committee (i.e., the KSO Approval Board) and four functional departments:

a. Quality Assurance Department

b. Configuration Management Department

c. Software Design and Development Department

d. Verification Department

Although this document is concerned primarily with the specification of KSO requirements (i.e., the "what to do" not "how to do it") the following procedures and/or mechanisms have already been established by

FACC:

a. Utilization of SRI's Hierarchical Design Methodology (HDM) for all formal specifications of security relevant software. This includes use of the SPECIfication and Assertion Language (SPECIAL) as the actual specification language.

b. Utilization of SRI's Boyer and Moore Theorem Prover for proving assertions utilizing HDM.

c. Utilization of Bell Laboratories' Programmer's Workbench (PWB)/UNIX system for actual software development and configuration management procedures. This in particular includes the Source Code Control System (SCCS) package supplied with PWB/UNIX.

d. Utilization of FACC-modified Modula compiler as originally developed by York University. This compiler operates under the PWB/UNIX.

FACC and the government has investigated considerable amount of time and dollars in perfecting these development and verification tools for KSOS development. It would not be prudent or cost effective to recommend other alternate mechanisms for subsequent KSOS support and maintenance.

The material which now follows will discuss in detail the requirements of the four KSO departments and the Approval Board.

5.1 QUALITY ASSURANCE DEPARTMENT

The Quality Assurance Department exists to guarantee to the government that its various military standards, directives, and interests are followed. This department's effect is felt over every facet of KSO operation. For example, all DR's and requested KSOS enhancements are reviewed by Quality Assurance; all readied KSOS enhancements and/or KSOS standard deliveries are approved by Quality Assurance.

The following duties are required of the Quality Assurance department:

a. Participating in the KSO Approval Board.

b. Providing direct interface to government accreditating agencies, performing in the role as KSO's security authority and advisor.

c. Approval of all final KSOS enhancement modifications, ensuring that all "regression testing" is initiated successfully within KSO established standards.

d. Approval of all final KSOS DR dispositions, ensuring that all "regression testing" is initiated successfully within KSO established standards.

e. Performance of KSOS site audits to ensure that KSO established standards of KSOS operation are being adhered to.

The Quality Assurance Department manager heads up the KSO Approval Board.

His participation, besides heading up the board, is to supply expertise in the area of quality assurance. The Approval Board, which contains members from all KSO departments, has overall responsibility for guaranteeing the basic charter established during KSOS's conception.

Current security policies and procedures will be the province of Quality Assurance. Here Quality Assurance will ensure that KSO and subsequent KSOS sites follow current government policies and procedures in the installation, maintenance and operation of KSOS-based applications. The importance of this responsibility cannot be over emphasized since total site accreditation depends on the proper security controls levied on KSOS-based applications. Specifically, Quality Assurance must guarantee to the government accreditating agency that all security relevant software is maintained, verified, installed and operated within strict procedures as accepted and understood by the accreditating agency and KSO. For example, for all DR processing involving security relevant software, Quality Assurance must ensure that appropriate reverification, if required, is initiated against both specifications and code. Quality Assurance must ensure that on-site training of KSOS operation meets the security standards of KSO. This involves both the disposition of security relevant software and its loading procedures. Finally, Quality Assurance is responsible for accreditating the "KSOS portion" of a KSOS-based site and if accepted will contact the designated government accreditating agency indicating KSO's approval.

Quality Assurance is also responsible for ensuring that all KSOS enhancement development follows established approved sequences. This involves the proper feasibility reviews by the KSO Approval Board, ensurance of appropriate verification cycles if required, software development and "regression test" sequences are initiated, etc. Finally, Quality Assurance guarantees that the KSOS software enhancements do indeed satisfy the baseline requirements as specified by the requesting KSOS site.

Discrepancy Report (DR) processing is very similar to the processing of KSOS enhancement requests. Normally, they both involve modifications and/or enhancements to KSOS software. Therefore, Quality Assurance must ensure that the appropriate testing is performed by the various KSO departments in DR processing. Since DR processing may involve security issues (i.e., if the DR is against security relevant software) Quality Assurance has the responsibility for deciding whether affected KSOS sites must disable portions of their system or, possibly, shutdown the entire system.

As a final responsibility, the KSO Quality Assurance Department must make periodic KSOS site audits to guarantee to the government that the site is continuing to follow proper KSOS operation procedures. This auidit checks for the proper disposition of security relevant software, proper DR processing, ensurance that site personnel who were originally trained by KSO are still involved with the operation. This KSOS site audit is similar to the security audits the government requires of all contractor

facilities which house classified material.

5.2 CONFIGURATION MANAGEMENT DEPARTMENT

The Configuration Management Department of KSO is a very important part of KSO and is responsible for the overall success of KSOS distribution to potential sites. Configuration Management works closely with the Quality Assurance Department of KSO in performing its duties. The following tasks are the responsibility of Configuration Management:

a. Control of all KSOS program source data.

b. Maintenance of all KSOS documentation.

c. Control of all KSOS software regression tests.

d. On-site training for KSOS sites.

e. KSOS installltion.

f. Discrepancy Report (DR) tracking.

As discussed earlier in this section, FACC has utilized certain source control procedures during the development of KSOS. Due to overall cost effectiveness, the use of these procedures should be continued by the KSO. The use of the Source Code Control System (SCCS) facility provided by PWB/UNIX is an excellent support tool for controlling KSOS source. The SCCS is a software facility designed to help large programming efforts control changes to the program source code. It provides facili-

ties for storing, modifying and retrieving all versions of any program or sub-program. It does this by identifying each software component by a "version number" and keeping with it information specifying who made the update, when it was made and why. The SCCS facility solves the following problems associated with the control of large software developments:

a. Alterations and/or updates to one version of a program sometime fail to get made to other versions.

b. When alterations and/or updates are made to a program, it is difficult to tell exactly what the change was or when and who made the change.

c. When multiple versions are supported in the field, it is difficult or next to impossible to generate an old version.

The SCCS facility in PWB/UNIX solves the above problems. The third problem mentioned above is extremely important and SCCS is capable of regenerating any program at any point plus maintain a complete history of the changes during the course of the project's life.

The documentation representing the use, design, and operation of KSOS is an importance component of the system in general. It will be the responsibility of the KSO Configuration Management Department to ensure the integrity of this documentation. KSOS documentation can be broken down into four groups:

a. Basic definition/specification documentation; i.e., A-specifications, B-specifications, etc.

b. Software maintenance documentation; i.e., C-specifications, regression tests, etc.

c. Verification documentation; i.e., formal specifications, "proof" documentation, etc.

d. User and training documentation; i.e., user manuals, presentations, etc.

Since some of this documentation currently exists as files under the KSOS PWB/UNIX developmental system, it seems prudent that all documentation be kept in this manner. PWB/UNIX's document generation facilities provide excellent flexibility in maintaining KSOS documentation.

Configuration Management will control all "official" regression testing of KSOS software. By "official" we are referring to the final tests made on a KSOS enhancement prior to insertion to KSOS systems. These tests are designed to thoroughly test out existing KSOS software, new software enhancements, and improvements reflecting DR rectification. Consequently, these tests must be controlled via the same mechanism as that used for overall KSOS source control, the SCCS facility of PWB/UNIX. It is important to stress here, that so often new enhancements are checked out adequately in the areas where the changes were made, however, other system pertubations are not usually encountered until the system is operational. Regression testing ensures that "other parts of the system"

still function correctly after the alteration has been made.

Configuration Management will be responsible for performing all on-site training of potential KSOS sites. This training will involve briefing site Security Officers (SO) on the impact of using a KSOS-based application, training the overall System Administrator (SA) in the correct disposition of security relevant portions of the system, and the training of site System Operators (SYO) in the operation of KSOS.

Closely related to on-site training of personnel, is the actual installation of KSOS. Here Configuration Management is responsible for final personnel training during installation and generally will stay involved until both KSO and the site believe operation can be successfully handled by the site itself.

As a final responsibility, the Configuration Management Department will have overall management control of DR processing. This department will register incoming DR's, assign priorities to them, assign appropriate KSO personnel for their rectification, and, finally, configure them into the KSOS system. Following DR resolution, Configuration Management will be responsible for distributing the altered KSOS systems to those sites affected by the DR(s) and providing on-site assistance to the sites requiring it. Typcally, DR's will be used to report all alleged problems against KSOS. Information attached or contained within the DR will be a description of the problem as evidenced by the symptoms, supporting data, and instructions to reproduce the problem. The DR will contain the following information:

a. DR number

b. KSOS site number

c. KSOS version number

d. date

e. name and phone number of person submitting DR

f. configuration description

g. DR information; i.e., problem situation and amplyfying information.

The above information will represent the initial DR entry. Subsequent inqueries, related DRs, will also be appended to this DR information entry. Following DR registration, Configuration Management will track the DR via this entry by appending pertinent information during its processing cycle. This information will include who in KSO is working on the DR, any required correspondence with the site for additional information, possible on-site study of the problem (if potentially configuration dependent), and the final resolution of the DR.

5.3 SOFTWARE DESIGN AND DEVELOPMENT DEPARTMENT

The Software Design and Development Department is responsible for the actual KSOS software. This department contains personnel with expertise in all facets of KSOS software. Therefore, it is clear that this depart-

ment is not only responsible for developing new enhancements, but is also responsible for rectifying DRs given to it by the Configuration Management Department. It is important to distinguish between this department and Configuration Management. Configuration Management has control of the correct disposition of KSOS program source code, of KSOS regression tests, and DR tracking. The Software Design and Development Department is responsible for the actual implementation of KSOS souce alterations, the design of regression tests, and the resolution of KSOS DRs.

An important aspect of this department is the development of special site-specific trusted software. Although KSO will be organized to allow sites to develop software locally, it is believed that for most situations, trusted software development will be accomplished by KSO. Even if a site has the capability for trusted software development, KSO has the final responsibility for final verification and subsequent accreditation of the software.

Trusted software development will involve the close coordination with the KSO Verification Department. Trusted software must first be formally specified by the Verification Department and this formal specification then becomes the design baseline by which the Software Design and Development Department uses for subsequent design and implementation. Following design of trusted software, this department then, in close concert with the Verification Department (who is now verifying the formal specifications), initiates software development and designs the set of

regression tests required to satisfactorily test out the software. Following successful implementation and testing of the trusted software, Quality Assurance then approves of the entire cycle prior to letting Configuration Management configure it into KSOS.

Since new candidate KSOS sites may have hardware devices not accommodated by the existing KSOS system, new device drivers will have to be developed. Therefore, the Software Design and Development Department must be cognizant of all new hardware release, both from Digital as well as leading independent suppliers, in order to anticipate the requirement for new KSOS device drivers. This is important since device drivers reside in the KSOS kernel and, therefore, must be carefully constructed and configured.

From the above discussion it is clear that personnel of this department must be well versed in using the implementation programming languages of KSOS development, i.e., C programming language and Modula, and have expertise in UNIX usage as a general time sharing operating system. Since KSOS emulates a formal version of UNIX (i.e., version 6), it is important that this department be cognizant of all formal UNIX releases and receptive to possible enhancements reflecting these new enhancements. It is important that the emulation package of KSOS reflect accurate versions of UNIX.

5.4 VERIFICATION DEPARTMENT

The KSO Verification Department is responsible for all verification of security relevant software. As discussed elsewhere in this document, security relevant software is composed of the KSOS kernel itself, the NKSRS package, and site-specific trusted software. All this software must be developed along precise lines allowing final verification as required for overall site application accreditation. The Verification Department is responsible for the following tasks leading up to system verification:

a. Production of formal specifications of all security relevant software.

b. Conducting proofs of formal specifications.

c. Correlation of formal specifications with "code instances."

d. Conducting "code proofs."

An important aspect of KSO verification is the "verification" procedures required for those KSOS components that are altered due to enhancement requests and/or DR resolution. In order to reduce cost impacts of reverification, it is important that the KSO Verification Department establish procedures for carefully "stratefying" security relevant software such that respecification is minimized. Consequently, this willl also reduce the number of new theorems that need be proved for reverification.

Since verification technology is going through a "renaissance" it is important that the Verification Department be cognizant of all technolo-

gical break throughs in this highly complicated science. Although it seems prudent at this time to continue using the established verification tools utilized for KSOS development, it may be wise for future development and support to expand on these tools and techniques to exploit new technologies.

5.5 KSO APPROVAL BOARD

The KSO Approval Board is responsible for ensuring the overall integrity of KSOS's charter during the ongoing support of KSOS. The board is composed of the managers from the four KSO departments and the KSO director. The Quality Assurance Department manager chairs the board. The following duties are the resp nsibility of the board:

a. Approval of all KSOS sites.

b. Approval of all KSOS enhancements.

c. Review of all incoming DRs.

d. Conduction of periodic audits with the controlling cognizant government agency.

The KSO Approval Board reviews and approves all KSOS site requests. In particular, for those sites requiring classified operations (as most sites), this board will confirm that the controlling government agency has indeed given its permission for KSOS application.

All requests for KSOS enhancements must be reviewed and approved by this board. This includes all requests for UNIX emulator enhancements and new trusted software. The purpose of this review cycle is to insure that the requests are well stated (i.e., understandable) and that they do not cause problems with the overall integrity and security of KSOS proper.

The Approval Board must review all incoming DRs to insure that they are not indicative of potential security violation situations. If the DR is deemed to be of importance from a security point of view, then KSO must evaluate the DR immediately and make a decision as to the true impact to site security. Possibly, this site, and all other affected sites, may have to be shut down until the DR can be rectified.

Since KSO will be part of the government's accreditation procedure, it must undergo periodic government audits. The KSO Approval Board will represent KSO for this audit. During this audit, all aspects of KSO operation will be reviewed by the government to insure that KSO is up-to-date in following government security policies and procedures.

SECTION 6

KSOS SITE RESPONSIBILITIES

Before a potential KSOS user site is approved for receipt of KSOS, a set of shared responsibilities must be established between that site and the KSO. Based on discussions in Section 4 on KSOS user site categorization and Section 5 on KSO responsibilities, it is clear that the set of responsibilities to be established are somewhat contingent on the site's category. This section will discuss all phases of site responsibilities from the so-called "simple" sites (i.e., those without special requirements and operate as an unclassified site) to the "complex" sites (i.e., those sites operating in a multi-level environment and having special requirements demanding KSOS enhancements). The following responsibilities for KSOS user sites are identified:

a. Application identification and overall government approval.

b. Specification of site categories; i.e., security requirements, special requirements, and site capabilities.

c. Identification of responsible site personnel.

d. Site/KSO training phase.

e. Discrepancy Reporting (DR) requirements.

f. On-site software modification control procedures.

Some of the above topics are required for approval of KSOS usage, while others are required during the use of KSOS. These are discussed in the following paragraphs.

6.1 APPLICATION IDENTIFICATION

Typically government sites which desire a KSOS-based system have a requirement for operating in a multi-level environment. For this typical situation the site has identified a specific application which, in close concert with KSOS, will satisfy the operational requirements of the project. In order to even entertain the possibly of operating in a multi-level environment, the government site must gain approval from both the agencies responsible for the classified information and the overall government agency responsible for site/application accreditation. Part of the overall site/application accreditation procedure is the utilization of KSOS. Therefore, it is mandatory that KSOS usage is tightly controlled by the KSO in order to comply with its part of overall site/application accreditation.

For those KSOS user sites which will operate in a unclassified mode (e.g., typically some university) application identification and accreditation is not a requirement. Although those sites desiring to operate is a "system high" classified mode must acquire overall site/application

accreditation (as any site which handles sensitive classified must do) this is not further complicated with the addition of KSOS. It is believed here that standard security policies controlling physical site security, personnel security clearances, sensitive information handling procedures, TEMPEST, etc. will satisfy site/application accreditation. Therefore, it is those sites which must operate in a multi-level environment that require special accreditation for KSOS utilization. It is the responsibility of all potential KSOS user sites categorized as such to gain approval from the cognizant agencies prior to formally requesting KSOS from KSO.

For those sites requiring special trusted software development, additional government approval cycles will be required. In this situation we are referring to that trusted software which may violate one or more of the inherent security principles of KSOS. Again, this approval must be acquired prior to submitting a formal request to KSO for KSOS.

6.2 SITE CATEGORIZATION

In Section 4 of this document a set of site categories were established and described which were considered important in establishing the KSO/site interplay. One of the key responsibilities of a potential KSOS user site is specifying these categories. These categories include:

a. site security

b. special site requirements

c. site capabilities

As was discussed in detail in the previous subsection, site security requirements, in regards to the operation of the KSOS-based application, is a key factor in determining the procedures for acquiring approval to use KSOS. It is the responsibility of the potential KSOS user site to specify to KSO the operational security requirements of the KSOS-based application. As discussed previously, this security requirement can range anywhere from none (i.e., unclassified operation) to significant security requirements (i.e., multi-level operation requiring special trusted software). The site security categorization will play heavily in determining KSO's priority in its support and maintenance role for a site.

A potential KSOS user site must specify precisely any special requirements it might require of KSOS. Here, we are referring to alterations and/or enhancements to the standard KSOS package as well as special trusted software requirements. As discussed in section 4 it will not be an uncommon occurrence for potential KSOS user sites to request enhancements to KSOS. Typically, these will be alterations to the UNIX emulation package supplied with KSOS. Less frequently occurring change requests would involve the Security Kernel and NKSRS components of KSOS. Hopefully, alterations to the Security Kernel would involve the inclusion of non-standard devices not supported by KSOS. Actual changes to the kernel itself would not be requested very often if ever. Changes to NKSRS

will normally be satisfied by new site-specific trusted software. Special trusted software requirements will be another requirement of sites that will occur quite frequently. This will probably prove true for sites operating existing UNIX-based applications that have embedded security relevant software. For example, a typical situation would be where a UNIX-based application is operating at "system high" and then a new requirement is imposed on it to operate in a multi-level environment. KSOS would then be utilized in place of the standard UNIX kernel. The application-specific software which is security relevant would then be either absorbed in existing NKSRS functions or redeveloped as trusted software. The latter would probably be the prevalent case.

It is important to note here that part of the site categorization and subsequent KSO approval is based on the hardware configuration the site desires to operate with. Typically, this is only of concern when one or more non-standard devices are present which may require new device drivers. However, it is KSO's responsibility to make sure that the hardware is not so non-standard that certain fundamental assumptions of KSOS are not circumvented by the hardware, hence making the system insecure. Obviously this covers special hardware modifications to the processor itself (i.e., memory management operation, micro-code alterations, etc.).

Finally, a potential KSOS user site must specify, in accordance with special site requirements, what portions,if any, of these requirements it plans to develop itself. This is referred to as site capabilities. As

discussed in section 4 KSO assumes that the site will develop the application software which will operate under KSOS, therefore, the site (by site we are referring to site personnel and/or site appointed contractors) will have a certain nucleus capability for developing software. Since it is important that site capabilities be exploited whenever possible, the KSO/site interplay must allow for site modifications to KSOS. Of course these modifications must be controlled carefully in order to maintain an overall chain of responsibility for security and system integrity. The procedures of incorporating site capabilities with special site requirements of KSOS will be controlled by a "contract" between the site and KSO. This agreement will specify responsibilities both for the secure operation and operational integrity of the system as a whole. Typically, as with any other service/maintenance agreement between a user/vendor, whoever makes the changes has the responsibility of these changes. Again, here typically site modifications to KSOS will involve those areas which are not security relevant. If the site plans to develop trusted software it must do so under the control and final accreditation of the KSO.

In summary, the potential KSOS user site must submit to KSO a statement of the security level its KSOS-based application is to operate under, a list of special requirements for its version of KSOS, and a commitment as to what special requirements it plans to develop itself.

6.3 SITE PERSONNEL

As part of the approval process for KSOS utilization, the potential KSOS user site must identify personnel who will have responsibility for KSOS operation. These responsibilities will include enforcing security procedures for KSOS operation, and the day in and day out operation of KSOS, including proper discrepancy reporting procedures. Of course, if a site is to utilize KSOS in an unclassified mode, then site personnel requirements are not necessarily required. In Section 4 we identified three site personnel categories that are required for the secure and successful operation of a KSOS-based application. These are:

a. Security Officer (SO)

b. System Administrator (SA)

c. System Operator (SYO)

As stated in Section 4 these identified personnel would normally already exis. on a site which housed computer systems and processed sensitive information. However, in order to operate a KSOS-based application, additional operational procedures must be established and complied to involving both security and system integrity. The role these personnel play in KSOS operation is described further in following subsections.

6.4 SITE PERSONNEL TRAINING

Upon KSO site approval for KSOS operation, the designated cognizant site personnel must go through an indoctrination and training phase prior to receiving KSOS. Again, typically these training courses are requirements of only those sites employing KSOS for classified information. This training phase will cover the following topics:

a. Ramifications of operating in a multi-level environment.

b. Secure handling procedures for KSOS.

c. Operating procedures for KSOS.

d. Discrepancy Reporting (DR) procedures.

e. Special software/KSOS interface training.

KSO will conduct a special seminar for the Security Officer (SO), the System Administrator (SA), and other designated site personnel (i.e., software engineers who may be developing special site-specific software) providing proper indoctrination in operating a multi-level system. In this seminar KSO will present the basic philosophy of security kernel technology, formal verification, and the utilization of trusted software. In general this seminar will cover the subtleties involved in formal security such that on-site personnel will have the proper appreciation and respect for KSOS-based applications.

Since one of the properties of a security kernel is protection from operational 'tampering', it goes without saying that protection of the actual physical medium it resides on is equally important. The KSO will specify a set of handling procedures involving the delivery of KSOS magnetic and paper mediums. This will include initial system deliveries, new system updates, and special DR updates. This training phase will be for the SO and the SA. Generally, the SA will have the overall responsibility for enforcing these handling procedures.

KSO will train designated System Operators (SYO) in the day in and day out operation of KSOS. This training will involve system initialization procedures, system recovery and file system backup procedures, and any other operations necessary to insure a secure and successful operation of a KSOS-based system. This training phase will be given to the SYOs and the controlling SA.

The SA will be responsible as a central site despository for reporting all discrepancies encountered while operating the KSOS-based system. KSO will provide the SA with a set of DR forms and train him in the procedures of filling out the DR's and the site/KSO communication protocol. Since potential flaws in a KSOS-based application may lead to the compromise of sensitive information, it is important that the DR procedure be well established. Although the SA has overall responsibility of discrepancy reporting to the KSO, the success or failure of this procedure is contingent upon the immediate compliance of application users. Therefore, part of this DR training will involve cognizant users of the

KSOS-based application.

If a site requires special modifications and/or enhancements to the KSOS system where one or more of these are to be developed by the site, then a special software training course customized for the situation will be provided by the KSO. Typically, this course will be given only to those analysts and software engineers involved with developing the enhancements. Since it is desirable to exploit on-site capabilities whenever possible, the complexity and depth of this special software course is open. For example, if site personnel, in order to maintain compatibility with existing UNIX-based applications, desire to add a set of new UNIX system calls to the UNIX emulator, KSO would have to provide training on the UNIX emulator/kernel interface.

6.5 DISCREPANCY REPORTING (DR)

As discussed in earlier subsections, discrepancy reporting is an especially important responsibility of the KSOS user site. Due to the special security sensitivity of most KSOS-based applications, DR procedures are extremely important. DRs will be used to report all potential system discrepancies. Information supplied by the DR will include but not be limited to, the following:

a. Detailed description of the problem, including circumstances leading to the problem.

b. Supplementary information such as terminal output and/or computer dump listings.

c. Organization information:

1. user organization

2. user name and telephone number

3. date problem occurred

This information will be submitted to the on-site System Administrator (SA) who will then analyze the DR. If deemed necessary, the SA shall submit an official DR to the KSO. The SA will include any other amplifying information to KSO; e.g., hardware configuration changes, etc. The SA will attach a priority on the DR based on the site's requirements. If the SA suspects that due to the nature of the DR, security relevant software may be in error, he must contact the Security Officer (SO) immediately and call the KSO explaining the problem. Based on this SA/KSO dialogue, the KSO and site SO will make a decision to either shut the system down immediately, disable certain functions of the system, or continue operation until more information is available. This last option will typically be taken when the KSO is convinced that the DR does not have security ramifications.

6.6 ON-SITE SOFTWARE MODIFICATION CONTROL

For those KSOS user sites which plan to develop enhancements to the delivered KSOS system, a set of procedures as established by the KSO must be followed. These procedures will have been established during the site/KSO dialogue leading to KSO approval for KSOS use. Typically, these procedures are tailored for the nature and extent of the enhancements under consideration. Again, these are dependent on the overall site categorization, including both the special site requirements and site capabilities.

The purpose of these procedures is two-fold:

a. To maintain a secure operation.

b. To maintain the operational integrity of the system.

The first purpose is concerned with the proper development, verification and installation of software enhancements which are relevant to security. As stated elsewhere, this would normally be the development of new trusted software. This situation is by far the most complicated procedure since it will involve intimate site/KSO interplay. Section 7 provides a detailed scenario of this interplay. The second purpose is important because new enhancements must operate correctly for overall system success. The procedures are directly contingent on the established responsibility of these system enhancements. Depending on the pre-established arrangements between the site and KSO, these procedures can be nil or quite comprehensive. If the alterations are not security

relevant and are for experimental purposes for that site, then KSO may claim no responsibility for their success/failure. On the other hand, the pre-established support role of KSO may include close supervision and the possible redistribution of these enhancements to other KSOS user sites. For the latter case, site/KSO interplay may become quite involved.

In summary it is clear that, depending on overall site categorization, site/KSO interplay is quite flexible. This interplay is based on a combination of required security controls and optional requirements and capabilities of the individual sites.

SECTION 7

KSO/KSOS SITE INTERPLAY

In the previous sections we defined the KSOS Support Organization (KSO) and the KSOS sites and their perspective responsibilities. In this section we present representative scenarios illustrating the interplay between the KSO and KSOS sites. The scenarios discussed below (depicted in Figures 7-1 through 7-5) illustrate the following situations:

a. Unclassified site requesting and receiving KSOS.

b. Classified site, operating system high, requesting and receiving KSOS.

c. Classified site, operating multi-level, requesting and receiving KSOS.

d. Special trusted software development.

e. KSOS site encountering DR in trusted software.

Before pursuing the above scenarios, the nomenclature used in Figures 7-1 through 7-5 will be explained. In each figure, required site actions are represented on the left and KSO actions are represented on the right. Each required action is indicated as a circle and labelled accordingly. The order of the actions is indicated by following the "arrowed" paths

connecting the circles. A particular capability of the site is specified on the top of the figure (e.g., unclassified site, software development capability, etc.).

7.1 UNCLASSIFIED

This scenario is quite simple, as shown in Figure 7-1. Here, the site submits a request to the KSO for a standard version of KSOS. We will assume that the site is a university interested in conducting research on security kernels. The standard version of KSOS would be the security kernel, UNIX emulator, and the standard NKSRS packages. It is important to note here that spatial site-specific trusted software would not be distributed for two reasons. One, it is not part of the standard package, and two, the actual existence and/or nature of the trusted software may be classified.

Following receipt of the KSOS request, KSO categorizes the site and establishes its priority for subsequent support and maintenance. For this example, we will assume that the site plans to experiment extensively with KSOS and that, therefore, KSO will not be responsible for any subsequent maintenance or support.

KSO then performs on-site training of interested personnel. Again, since the site is not operating KSOS in a classified environment and only wants KSOS for research, on-site training is a relatively simple phase of this request.

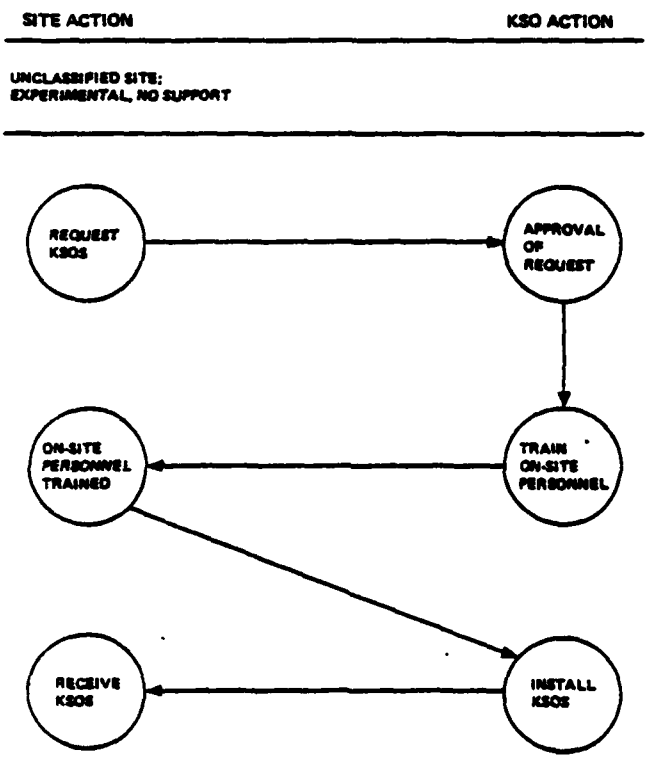


Figure 7-1. Unclassified Site

Finally, KSO installs KSOS at the site and answers any other questions the site may have. At this time the site has complete responsibility for its KSOS system, having not requested follow-on support and maintenance from KSO.

7.2 CLASSIFIED SITE OPERATING SYSTEM HIGH

This scenario is considerably more complicated than the previous scenario just presented (see Figure 7-2). Here we have a site which will operate in a classified system high environment. Typically, this site desires KSOS because it will in the future progress to a multi-level environment, a situation demanding KSOS's guaranteed operation.

First, the site must gain overall approval for its planned application. This approval must come from both the organization held responsible for the classified information going through the system and the agency which will ultimately accredit the operation.

The site must then request KSOS from the KSO, supplying the necessary information required for site categorization. KSO, upon receiving the site request, then categorizes the site and establishes a priority for future support and maintenance functions . Part of this categorization phase involves KSO confirming overall approval for KSOS usage in a classified environment.

KSO next analyzes required customizations of the KSOS package. For this scenario, we will assume that the site desires enhancements to the UNIX

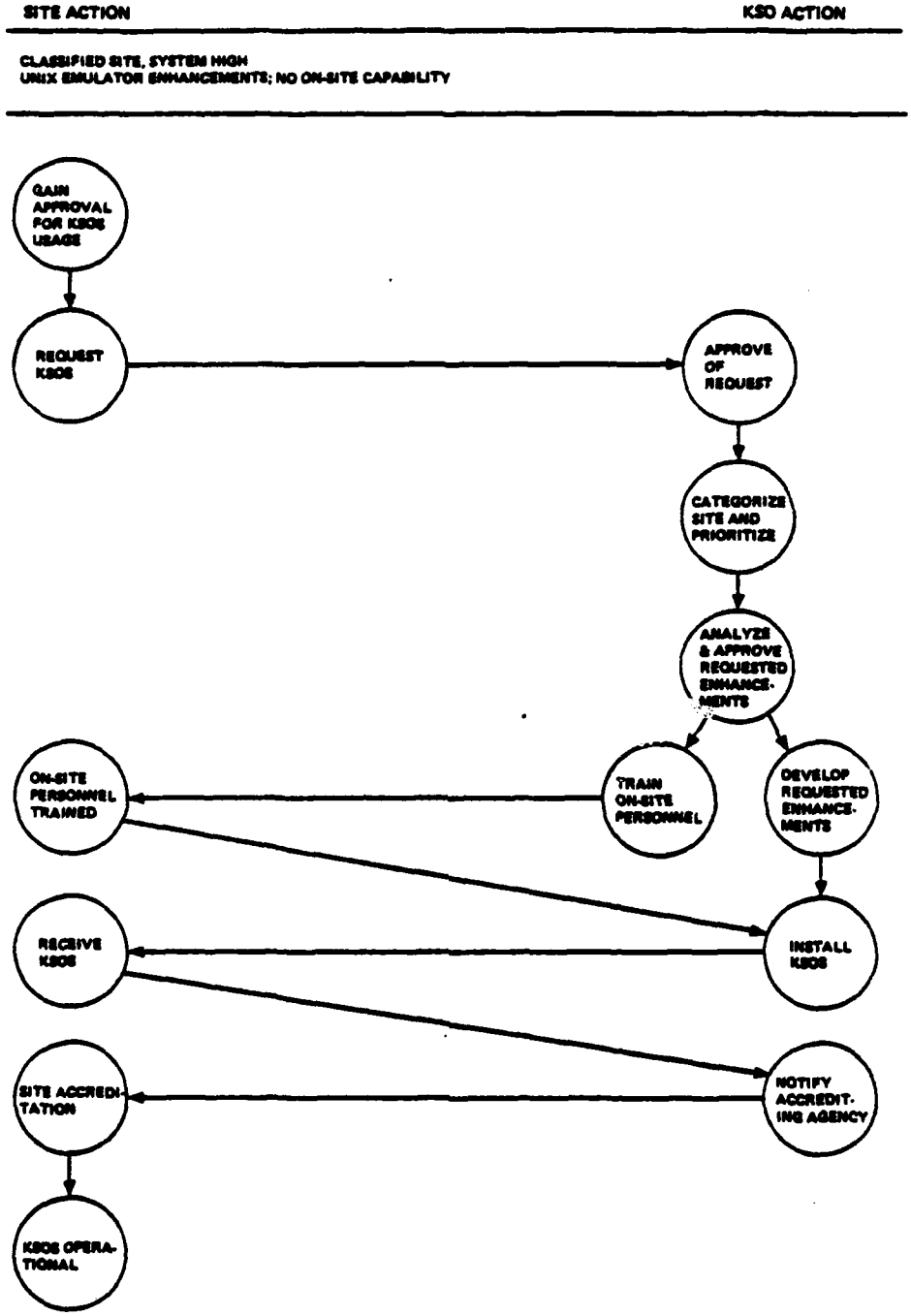


Figure 7-2. Classified Site Operating System High

emulation package (e.g., new system calls). KSO then approves of the enhancements as being sound and implementable.

Since the requesting site will be operating in a classified environment, KSO must conduct extensive training sequences with designated on-site personnel. This training will include security, DR submissions, and overall system operation and integrity. This training will commence somewhat in parallel with KSOS customization development sequences.

KSO then proceeds to develop the requested enhancements. Since the requested enhancements are not security relevant, special verification/reverification phases will not be required. Following development and complete checkout of the new enhancements, KSO's quality assurance department approves of their release to the configuration management portion of KSO.

KSO then installs the enhanced KSOS system at the site and supervises the on-site personnel in last minute details for KSOS operation. Then KSO notifies the responsible accrediting agency that KSOS has been installed and on-site personnel are qualified for its operation.

Now the accrediting agency proceeds with the remainder of application accreditation. Following accreditation, the site can then operate its KSOS-based application at the classification level approved by the accreditating agency.

7.3 CLASSIFIED SITE OPERATING MULTI-LEVEL

Figure 7-3 illustrates the complicated scenario for multi-level operation. Here the site requesting KSOS will operate in a multi-level environment and require special trusted software for its application. In this scenario we will also assume that the site has no special on-site capability for producing trusted software.

As with the previous scenario, the site must gain overall approval from the accrediting agency in using KSOS in its application. This particular approval cycle can be quite involved since the site is operating in a multi-level environment and requires the use of special trusted software. Depending on the nature of this trusted software, the approval cycle may be quite complicated. Following approval for KSOS use, the site makes an offical request to KSO. As with the other scenarios, KSO analyzes the request and categorizes the site accordingly. Again, KSO must insure that the site does indeed have the proper approval for utilizing KSOS in its application.

Next KSO interfaces with the site to obtain the precise operation requirements of the special trusted software. KSO must then approve the soundness (with respect to security as well as integrity) of the trusted software prior to any subsequent development. Since the site has no capability for producing this type of software, KSO will be responsible for its entire development. Trusted software development is a non-trivial task (see paragraph 7.4 for a detailed scenario) involving formal specifications and verification. Following successful development of the

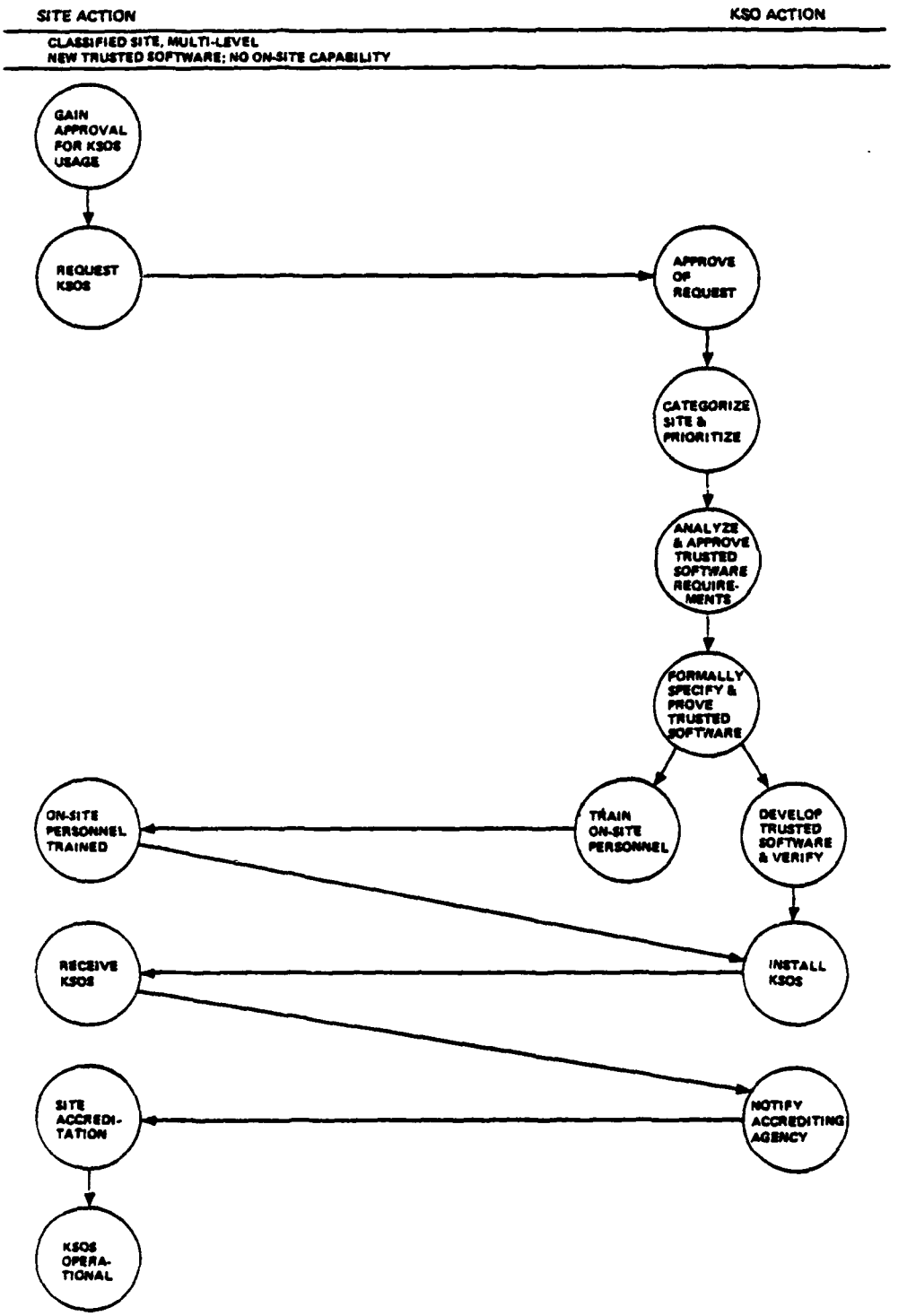


Figure 7-3. Classified Site Operating Multi-Level

trusted software KSO must then pass it through its Quality Assurance Department prior to release and installation by its Configuration Management Department.

Parallel to the special trusted software development, KSO must train the designated on-site personnel in KSOS operation. As with the previous scenario this training covers overall security matters, DR sequences, and the operation of KSOS.

Following training and trusted software development, KSO installs KSOS at the site. Upon successful installation, KSO notifies the appropriate agency that KSOS has been installed and the site is qualified to operate it securely. This completes the KSOS-phase of overall site/application accreditation.

7.4 TRUSTED SOFTWARE DEVELOPMENT

Figure 7-4 illustrates the scenario for trusted software development. Here we assume that the site has software capability and, therefore, is competent in generating good software design and code. The KSO will have the responsibility of overall approval and verification of the software.

In this scenario the site must come up with the 'English' specifications of the trusted software. This includes the operational requirements of the trusted software, including any potential circumvention of KSOS's security principles. If this software is to interface with users or untrusted software or both, these interfaces must be precisely specified.

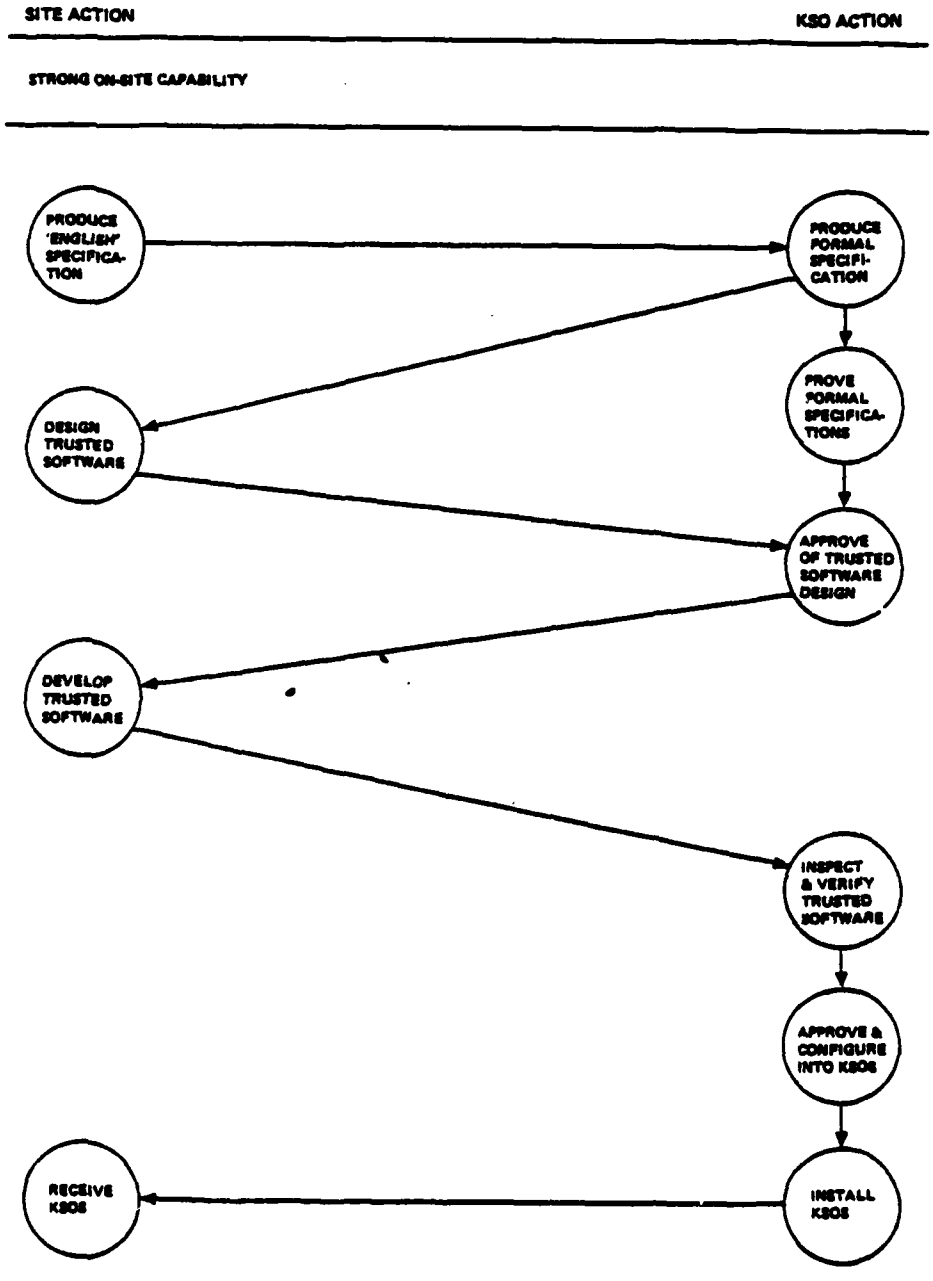


Figure 7-4. Trusted Software Development

This "English" specification is then analyzed by KSO to insure that overall requirements are sound and that they do not involve too much complexity such that verification is impossible. This phase of development will normally involve a lot of site/KSO interplay in order to "hone" down the trusted software requirements.

Following general agreement on the "English" specifications (probably now modified), KSO will produce a set of formal specifications of the trusted software operation. This formal specification will form the baseline by which the software will be developed.

Following receipt of thh formal specifications, both the site and the KSO perform two tasks in parallel. Since the site has local software capability, it will start to design the trusted software based on the formal specification. KSO will commence proving the formal specifications. This proof will guarantee that the formal specifications comply to the operational requirements of the trusted software. Possibly, this proof cycle will require alterations to the formal specifications. If this happens, the changes will be indicated to site personnel producing the trusted software design.

Following successful proofs of the formal specifications KSO will analyze and approve of the site trusted software design. This phase will ensure that the design does indeed follow the formal specifications. Following an intensive dialogue between the site and KSO, the design will be agreed to by both and frozen. Then the site can start actual software development. In order to accomplish its development, KSO will furnish the site

a "develop and test" version of KSOS for trusted software development.

After the trusted software has been developed and tested by site personnel, KSO will take "custody" of the software and inspect it for compliance to the formal specifications. Depending on the sensitivity of the application and what privileges the trusted software may have, actual "code proofs" may be required for complete verification. After KSO has successfully verified that the trusted software does operate correctly and "to the spec", it is passed through its Quality Assurance Department. Following approval of quality assurance (which will verify that all integrity and verification test phases do comply to KSO standards), KSO's Configuration Management Department is given the new trusted software for insertion into KSOS.

7.5 TRUSTED SOFTWARE DISCREPANCY REPORTING (DR)

This final scenario illustrates KSO/site interplay for DR sequences involving trusted software (see Figure 7-5). Here we assume the site/application has been accredited and the KSOS-based application has been operation for some period of time.

Let's assume that site personnel detect that a particular trusted software package does not seem to be operating correctly. This discrepancy is immediately reported to the site System Administrator (SA) who, based on the potential severity of the DR, contacts the site Security Officer (SO). The SA also contacts KSO immediately describing the DR. Note this entire sequence has involved the telephone, not formal

SITE ACTION	K80 ACTION
CLASSIFIED SITE, MULTI-LEVEL WITH SPECIAL TRUSTED SOFTWARE	

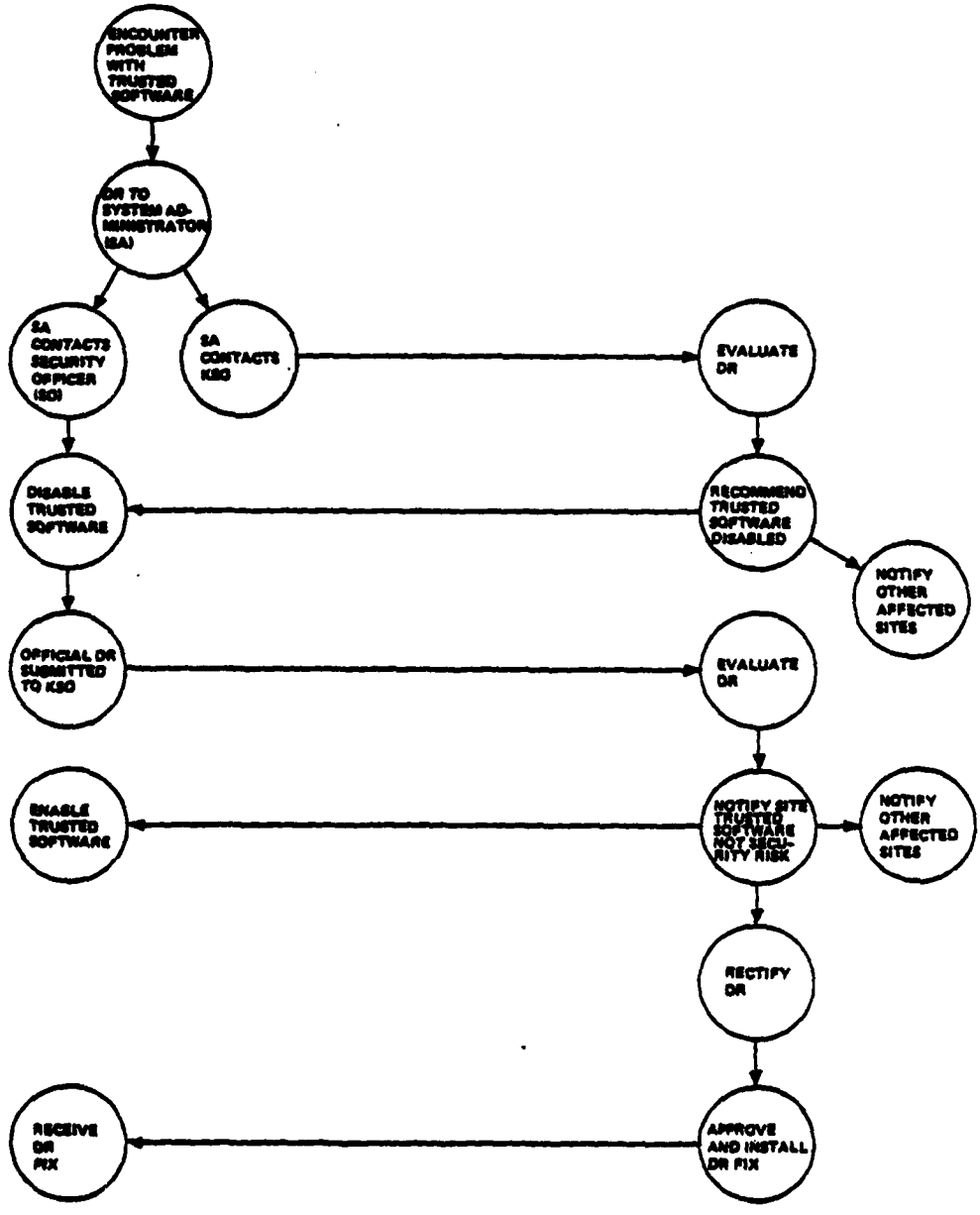


Figure 7-5. Trusted Software Discrepancy Reporting (DR)

paper. Due to the potential security ramifications of this DR, it is important that its resolution with respect to security be solved as soon as possible.

In this scenario, the KSO indicates to the site that the DR as described, has possible security ramifications and that that portion of the application be disabled until the DR can be studied further. The SA (under approval of the site SO) disables the trusted software immediately. Then an official DR is submitted to the KSO via normal channels.

In parallel with the above site action, KSO immediately notifies any other KSOS sites which may be affected by this DR (albeit, in most cases trusted software is site specific).

Upon receipt of the DR, KSO factors the DR into the 'DR chain' based on the overall priority assigned the site. When this DR is analyzed by KSO personnel, it is discovered that the discrepancy does not affect security relevant portions of the trusted software. KSO then immediately contacts all affected sites informing them that they do not have a security problem with the malfunctioning software. KSO then proceeds to rectify the DR, pass it through Quality Assurance and, finally, to the Configuration Management Department for insertion and delivery to the affected sites.